   
  
  
  
Phase I – Software requirements specifications-SRS Final

**Course Name:** Software Engineering

**Project:** Social Network

**Professor:** Dr/Mohammed El-Ramly

**Teacher Assistant:** Eng/Mohamed Samir

**Leader Name:** Aya sharaf EL-den

**Leader ID:** 20120124  
**Leader Email:** Ayasharaf25@gmail.com

Contents

[Team 2](#_Toc403244345)

[Document Purpose and Audience 2](#_Toc403244346)

[Introduction 3](#_Toc403244347)

[Software Purpose 3](#_Toc403244348)

[Software Scope 3](#_Toc403244349)

[Definition 3](#_Toc403244350)

[Requirements 4](#_Toc403244351)

[Functional Requirements 4](#_Toc403244352)

[Non-Functional Requirements 6](#_Toc403244353)

[Non-Functional Requirements 11](#_Toc403244354)

[System model 12](#_Toc403244355)

[Use Case Tables 12](#_Toc403244356)

[Ownership Report 26](#_Toc403244357)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20120124 | Aya Sharaf El-den | [Ayasharaf25@gmail.com](mailto:Ayasharaf25@gmail.com) | 01127811311 |
| 20120003 | Abrar Mo’men | abrar.moemen@gmail.com |  |
| 20120096 | Assmaa Khaled | Assmaa-khaled-fci@hotmail.com |  |

# Document Purpose and Audience

The purpose of this document is to explain the system Requirements specifications of the SOCIAL NETWORK.   
It will explain also how the system will deal with the user and his/her actions

# Introduction

## Software Purpose

* This software is a simulation to Facebook, as social networks is important to communicate with each other’s.
* Support user to create groups and socialize

## Software Scope

Our software is about a social network that aim to collect people together with some feature to make this collect much easier like sending message, make groups and pages and add friends.

Component of our software: Users, Groups, Posts, pages, hashtag, calendar and Events.

Our software can Sign up for an account, Sign in, add Friend request, Make a group or page and Events

Users: normal, verified.  
Groups: the user can create and join the group also can send message to the group or post on it.

Pages: the user can create and like the page also can post on it if the user is verified he/she can view number of likes and list of users.

## Definition

Social network is the software that will help the users to communicate easily with some features to be in touch in many ways like events, groups or Pages.

# Requirements

## Functional Requirements

* + 1. Users.
       1. Create account.
       2. Log in.
       3. Edit Info.
       4. Remove data.
       5. Send friend request.
       6. Accept request.
       7. Remove friends
    2. Posts.
       1. Write post.
       2. Edit post.
       3. Delete post.
       4. Edit privacy.
       5. Share post.
       6. Comment on post.
       7. Like post.
    3. Groups.
       1. Create group.
       2. Edit Info.
       3. Join.
       4. Leave.
       5. Add people.
       6. Remove people.
    4. Pages.
       1. Create page.
       2. Edit Info.
       3. Add post.
       4. Invite people.
       5. Block people.
       6. Like page.
       7. Like post.
       8. Share post.
       9. View number of likes.
       10. View list of users.
    5. Messages.
       1. Send message to a friend.
       2. Send message to a group.
       3. Send message to page.
    6. Hash tag.
       1. View.
       2. Arrange by date.
       3. Arrange by importance.
    7. Calendar and events.
       1. View national festivals.
       2. Add reminder.
       3. Create event.
       4. Edit description.
       5. Cancel.
       6. Add post.
       7. Invite friends.
       8. Join.
       9. May go.
       10. Leave.
       11. View number of attendees.
  1. Non-functional requirements.
     1. Usability.
     2. Maintainability.
     3. Reliability.
     4. Data integrity.
     5. Capacity.
     6. Implementation.

**Requirements.**

## Non-Functional Requirements

**2.1.1-Usesrs:**

There are 2 types of users: Normal user and Verified user. All users can create their own accounts and then log in to those accounts. They can edit information on their Information page and also remove data. They can add and remove friends.

**2.1.1.1- Create account:**

To use the social network every user has to create his own account with his personal information (Name, Email, Password, and Date of birth, Gender, Education, Phone number, Type-Normal or Verified user-).

**2.1.1.2- Log in:**

To start interaction with the social network each user has to log in to account he has created by writing his email and password then by clicking on Log in button.

**2.1.1.3- Edit Info:**

The user can edit his information after creating the account by clicking the Edit Info button.

**2.1.1.4- Remove data:**

Every user can delete any piece of information he entered when he created his account.

**2.1.1.5- Send friend request:**

Each user can send friend request to another user to add him as a friend to his friends' list.

**2.1.1.6- Accept request:**

The user can accept the friend request which was sent to him to confirm that he wants the other user to be added to his friends' list.

**2.1.1.7- Remove friends:**

Each user can delete anyone from his friends' list.

**2.1.2-Posts:**

There are 2 types of posts: Normal post which is written by the normal user and verified post which is written by the verified user.

The normal post can only be seen by friends of the user but the verified post can be seen by anyone if the user wants.

Every user can write, edit and delete posts. They also can edit the privacy of the post.

Every user can share other user post to his profile page.

Each user can like the post of other user and write a comment on it.

**2.1.2.1- Write post:**

Each user can write post on his profile page which can be seen by others according to his type.

**2.1.2.2- Edit post:**

The user can edit his post after writing and sharing it.

**2.1.2.3- Delete post:**

The user can delete the post he wrote if he wants to remove it.

**2.1.2.4- Edit privacy:**

The user can edit the privacy of his post to change it from public to private or from private to public.

**2.1.2.5- Share post:**

Any post can be shared by the friends of its writer to their profile.

**2.1.2.6- Comment on post:**

The friends of the user can write comments on his posts.

**2.1.2.7- Like post:**

The friends of the user can like his posts.

**2.1.3-Groups:**

There are 3 types of groups: Public, Closed and Secrete.

Each user can create group and edit the information. Each user can join or leave a group if it’s public or closed group. All users can add or remove other users to the group.

**2.1.3.1- Create group:**

Any user can create a group to discuss anything with people by choosing group name and type.

**2.1.3.2- Edit info:**

The creator of the group can change the description of the group and edit its privacy.

**2.1.3.3- Join:**

Each user can join any group he wants if the group privacy was public or closed.

**2.1.3.4- Leave:**

Each user can leave the group if he found that he is not interested in the content and the group goal.

**2.1.3.5- Add people:**

Any member in the group can invite his friends to join the group to interact with him and with all of the group members.

**2.1.3.6- Remove people:**

The admin of the group can remove any member from the group if he neglects the group rules.

**2.1.4-Pages:**

Every page has a list of users: Admin, Fan.

Page admin can create page, edit the information, add posts to the page, block fan and invite people to like the page.

Any user can like the page, like post on the page, comment on a post, share post and invite his friends to like the page.

All users can view the number of likes and the list of users.

**2.1.4.1- Create page:**

Any user can create his own page which can represent product or actor or singer.

**2.1.4.2- Edit info:**

The page creator can edit the information and the description of the page whenever he wants to do.

**2.1.4.3- Add post:**

The page creator can add posts to his page.

**2.1.4.4- Invite people:**

Anyone who likes the page can invite all of his friends to like the same page.

**2.1.4.5- Block people:**

The page creator can block any user who neglects the rules of the page.

**2.1.4.6- Like page:**

Any user who has account on this social network can like any page on it.

**2.1.4.7- Like post:**

Any user who already likes the page can also like any post on the page.

**2.1.4.8- Share post:**

Any user who already likes the page can also share any post from the page to his profile.

**2.1.4.9- View number of likes:**

Any user can see how many users have liked this page.

**2.1.4.10- View list of users:**

The page admin can view list which contains the other administrators if exist and also he can view the list of likes.

**2.1.5-Messages:**

Each user can send message to another user, page or group.

**2.1.5.1- Send message to a friend:**

Any user who has an account can send messages to his friends to chat with them.

**2.1.5.2- Send message to a group:**

Any user who has an account and join a group can send message to the group admin.

**2.1.5.3- Send message to page:**

Any user who has an account and likes page can send message to this page and the admin of the page will see it.

**2.1.6-Hash tag:**

If the user clicked on any has tag, he will get all the posts which were written with this hash tag arranged by the importance of the post according to number of likes and by date.

**2.1.6.1- View:**

Any user can view all the posts that include a hash tag while clicking on this hash tag in any post.

**2.1.6.2- Arrange by date:**

The user can choose to arrange the posts contain a hash tag by date by clicking arrange by date.

**2.1.6.3- Arrange by importance:**

The user can choose to arrange the posts contain a hash tag by importance according to the number of likes on the post by clicking arrange by importance.

**2.1.7-Calendar and events:**

Each user has a calendar reminds him with national festivals, and then he can add any reminder to his calendar on the date he want.

Each user can create an event, edit the description and the date of this event, delete the event, write a post on the event page and invite friends to join this event.

Each user can choose to join, leave or may go to the event.

Each user can view the number of people attending the event.

**2.1.7.1- View national festivals:**

The user can view all national festivals which is initially declared in the calendar.

**2.1.7.2- Add reminder:**

The user can add something to be reminded with and give it a description.

**2.1.7.3- Create events:**

Each user can create an event in a specific time and give it a description and start and end time.

**2.1.7.4- Edit description:**

The creator of the event can edit its description or change the start and end time after creating it.

**2.1.7.5- Cancel:**

The creator of the event can cancel and delete the event after sharing it.

**2.1.7.6- Add post:**

Each user can add post to the event whatever he was the creator of the event or any other person.

**2.1.7.7- Invite friends:**

Anyone can invite his friends to join the event.

**2.1.7.8- Join:**

Any invited user can choose to join the event and interact with it.

**2.1.7.9- May go:**

Any invited user can choose that he may join the event.

**2.1.7.10- Leave:**

Any user joined the event can choose to leave it at anytime he wants.

**2.1.7.11- View number of attendees:**

Any user can see how many users have joined the event.

## Non-Functional Requirements

**2.2.1- Usability:**

This API contains the core functions of any social media site. Using these functions, developers can create their own social networks.

**2.2.2- Maintainability:**

This API is maintainable as any developer can edit, add or remove components while creating his own social network.

**2.2.3 – Reliability:**

This API is considered the basic library for any social network. It has the ability to adapt to be any social media website like (FB, Twitter, Acadox, …).

**2.2.4- Data integrity:**

The accuracy and consistency of the API is maintained and assures over its entire life cycle.

**2.2.5- Capacity:**

This API has the capacity to accept any additional components or functions which the developer of the system would add.

**2.2.6- Implementation:**

Programming language: This API must be implemented in java.

# System model

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-1.1 | |
| Use Case Name: | Create account | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | - | |
| Post-conditions: | The system adds that information to the database | |
| Flow of events: | **User Action** | **System Action** |
| 1- User chooses a username and a password. |  |
|  | 2- System checks if username is available. |
| Exceptions: | Alert error if that username is already exists. | |
| Includes: | The user should have an e-mail to Sign up for a new account | |
| Notes and Issues: | The user must write his/her password 2 times to confirm it | |

**2.1-1. Users.**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-1.2 | |
| Use Case Name: | Log in | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | The system Send A confirmation mail to the user to Sign in | |
| Post-conditions: | Open the profile of that user | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters the username and a password. |  |
|  | 2- System checks if username and password are correct. |
| Exceptions: | If wrong user can re-enter username and password | |
| Includes: | The user must be signed up successfully | |
| Notes and Issues: | If the user enter the password 3 times wrong the system will close the account | |

**2.1-2. Posts**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-2.1 | |
| Use Case Name: | Write post | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Log in | |
| Post-conditions: | Friends can see the post | |
| Flow of events: | **User Action** | **System Action** |
| 1- User writes the post and chooses privacy. |  |
|  | The system add the post to the profile of the user |
|  |  |
| Exceptions: | If the user Press “Cancel” or “ Back” nothing will happen | |
| Includes: | The user should have an active account | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-2.3 | |
| Use Case Name: | Delete post | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Write post | |
| Post-conditions: | Friends cannot see deleted posts | |
| Flow of events: | **User Action** | **System Action** |
| 1- User deletes an existing post. |  |
|  | System checks if he can do this operation |
| Includes: | the user wrote a post | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-2.4 | |
| Use Case Name: | Edit privacy | |
| Actors: | Developer, Normal and verified users | |
| Pre-conditions: | Write post | |
| Post-conditions: | How to post will appear to the user’s friends and in newfeeds | |
| Flow of events: | **User Action** | **System Action** |
| 1- User writes the post and chooses privacy. |  |
|  | The system updated the privacy of the post |
| Exceptions: | Only the user that wrote the post can edit the privacy | |
| Includes: | The post had been written | |

**2.1-4. Pages.**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-4.1 | |
| Use Case Name: | Create page. | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Log in | |
| Post-conditions: | People see that page and can like it | |
| Flow of events: | **User Action** | **System Action** |
| 1- User creates page for a particular product. |  |
|  | Page added to the system |
| Exceptions: | If another page has the same name he can merge | |
| Includes: | The user should’ve an active account | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-4.3 | |
| Use Case Name: | Add post to a page | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Create page | |
| Post-conditions: | The post added to the page and any one in the page can see it | |
| Flow of events: | **User Action** | **System Action** |
| 1- User writes the post and chooses privacy. |  |
|  | The system add it to the page |
| Includes: | The page is already created | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-4.4 | |
| Use Case Name: | Invite people to like the page | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Create page | |
| Post-conditions: | Number of likes of the page increases | |
| Flow of events: | **User Action** | **System Action** |
| 1 -user invites his/her friends to like his page |  |
|  | 2-the system send a msg to the user that someone invite him to page |
| Exceptions: | Cannot invite people who are already like this page | |
| Includes: | The page is already created | |
| Notes and Issues: | Cannot invite any one twice | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-4.10 | |
| Use Case Name: | View list of users | |
| Actors: | Verified users. | |
| Pre-conditions: | Create page | |
| Post-conditions: | -- | |
| Flow of events: | **User Action** | **System Action** |
| 1- User can view who likes his page |  |
|  | View list of users |
|  |  |
| Exceptions: | Only the user who created that page can lists the fans | |
| Includes: | The page is already created | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1-4.9 | |
| Use Case Name: | View number of likes | |
| Actors: | Verified users. | |
| Pre-conditions: | Create page | |
| Post-conditions: | List the fans in the page | |
| Flow of events: | **User Action** | **System Action** |
| 1- User views the number of likes on his page |  |
|  | 2-view the number |
| Exceptions: | No one can use this feature only verified accounts | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1-1 | |
| Use Case Name: | CreateGroup | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | Log in | |
| Post-conditions: | The group is already created | |
| Flow of events: | **User Action** | **System Action** |
| User writes the name of the group |  |
| Choose the privacy |  |
| Add other users | Add group details to the system |
| Exceptions: | Who created the group will be the ADMIN | |

**2.1-3. Groups**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1-2 | |
| Use Case Name: | join | |
| Actors: | Normal and verified users. | |
| Pre-conditions: | The group is already created | |
| Post-conditions: | The member joined the group | |
| Flow of events: | **User Action** | **System Action** |
| User writes the name of the group | If it’s already excites and the privacy is public |
| The user press join |  |
|  | The user added to the group |
| Exceptions: | If the privacy not public the user will need approval to accept his/her request to join | |
| Includes: | The group already excites | |

**2.1-5. Messages.**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.5.1 | |
| Use Case Name: | SendMsgs | |
| Actors: | Users who have account on the social network (Normal User) | |
| Pre-conditions: | Press Like to the page OR Join to the Group | |
| Post-conditions: | Starting write the certain Msg | |
| Flow of events: | **User Action** | **System Action** |
| Click on Msg Button | Open the Page that the user will write the Msg in it |
| Write the Msg and After finishing click SEND | Send the msg to the page or group |
|  | Showing a confirmation Alert that the msg successfully Sent |
| Click OK. |  |
| Exceptions: | If the User Click on cancel , The system won’t Send the msg | |
| Includes: | The user should like or join the group | |
| Notes and Issues: | If there’s any problem while processing this function the system will ask the user to re-send the Msg. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.5.1 | |
| Use Case Name: | SendMsgToFriend | |
| Actors: | Users who have account on the social network (Normal User) | |
| Pre-conditions: | Open The Receiver “ Friend” Profile | |
| Post-conditions: | Starting write the certain Msg | |
| Flow of events: | **User Action** | **System Action** |
| Click on Msg Button | Open the Page that the user will write the Msg in it |
| Write the Msg and After finishing click SEND | Send the msg to the receiver “ Friend” |
|  | Showing a confirmation Alert that the msg successfully Sent |
| Click OK. |  |
| Exceptions: | If the User Click on cancel , The system won’t Send the msg | |
| Includes: | The User Should be In The Receiver’s “ Friend” List | |
| Notes and Issues: | If there’s any problem while processing this function the system will ask the user to re-send the Msg. | |

**2.1.6-Hash tag:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.6.1 | |
| Use Case Name: | ViewRelatedHashTags | |
| Actors: | Normal Users and Verified Users | |
| Pre-conditions: | The user activate his/her account and clicking on the Hashtag | |
| Post-conditions: | List all posts that contains the hashtag | |
| Flow of events: | **User Action** | **System Action** |
| Click on the Hashtag | Open the Page that list all the posts contains the hashtag |
| Exceptions: | If the User Press “Back” , Nothing will appear | |
| Includes: | The User should have an active account | |
| Notes and Issues: | If the Hashtag in post which its privacy is PRIVATE wont appear in the list | |

**2.1.7-Calendar and events:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.7.1 | |
| Use Case Name: | AddRemainder | |
| Actors: | Normal Users and Verified Users | |
| Pre-conditions: | The user activate his/her account and Press “ Calendar” | |
| Post-conditions: | View the calendar with the Remainder | |
| Flow of events: | **User Action** | **System Action** |
| Click on “Calendar” | Open the Page that show the Calendar |
|  | Select a certain day | Mark this day as in that day the system will Remain the user |
|  |  | Appear an Alert to the user to confirm that the day is successfully selected |
|  | Press Done |  |
| Exceptions: | If the User Press “Back” , Nothing will Happen | |
| Includes: | The User should have an active account | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.7.2 | |
| Use Case Name: | CreateEvent | |
| Actors: | Normal Users and Verified Users | |
| Pre-conditions: | The user activate his/her account and clicking on the Calendar | |
| Post-conditions: | Show the Calendar elements | |
| Flow of events: | **User Action** | **System Action** |
| Click on Calendar | Open the Page that list all the Calendar’s Element |
| Exceptions: | If the User Press “Back” , Nothing will appear | |
| Includes: | The User should have an active account | |
| Notes and Issues: | If the Hashtag in post which its privacy is PRIVATE wont appear in the list | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2.1.7.8 | |
| Use Case Name: | JoinEvent | |
| Actors: | Normal Users and Verified Users | |
| Pre-conditions: | The user activate his/her account and view the Event’s Page | |
| Post-conditions: | Show the Event Description | |
| Flow of events: | **User Action** | **System Action** |
| Click on Join | Add the user add a new member in the Event |
| Exceptions: | If the User Press “Back” , Nothing will happen | |
| Includes: | The User should have an active account | |
| Notes and Issues: | The user can cancel his/her join | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Introduction | *Assma Khaled* |
| Requirements (functional – non-functional) | *Abrar Mo’men* |
| System Models | *Aya sharaf – Assma Khaled* |